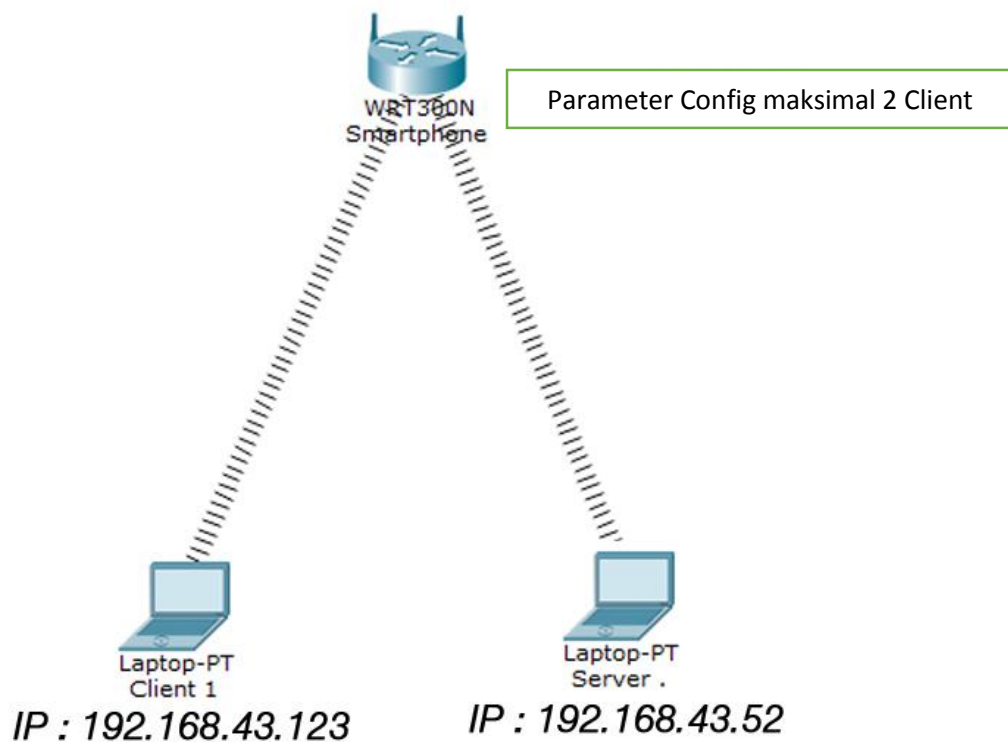


CARA MENJALANKAN PROGRAM

3.5.1 Diagram Jaringan Implementasi *MQTT*

Berikut merupakan gambaran diagram jaringan yang akan diimplementasikan. Yang terdiri dari 2 buah laptop dan 1 buah *smartphone*.



Gambar 3.5.1 Diagram Jaringan Implementasi MQTT

- *Laptop 1* : Berfungsi sebagai server *MQTT(Broker)* yang digunakan untuk *publish* dan *subscribe*.
- *Laptop 2* : Berfungsi sebagai client untuk men-*subscribe publish* dari server.
- *Smartphone* : Berfungsi sebagai *client* dan *access point*. Untuk mendapatkan akses ke jaringan internet agar saling terhubung satu sama lain. Dan untuk pengalamatan *IP* menggunakan

DHCP Server yang sudah ada di *device android* dimana protokol yang memudahkan mengalokasikan alamat *IP* pada suatu jaringan dengan *IP* dinamis , yaitu alamat *IP* terisi secara otomatis

3.5.2 Pengujian Koneksi Jaringan

Sebelum memulai tahap pengujian , yaitu melihat alamat *Host* yang tersedia pada *server* dengan cara ketik “*ifconfig*” maka akan muncul *inet address* ,*broadcast* ,dan *subnetmask* .

```
wlan0    Link encap:Ethernet  HWaddr 08:3e:8e:a6:86:ad
          inet addr:192.168.43.52  Bcast:192.168.43.255  Mask:255.255.255.0
          inet6 addr: fe80::a3e:8eff:fea6:86ad/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:80 errors:0 dropped:0 overruns:0 frame:0
          TX packets:54 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:12019 (12.0 KB)  TX bytes:10218 (10.2 KB)
```

Gambar 3.5.2 Koneksi Server

1. Menyambungkan *Laptop* dengan *Smartphone* sebagai *Access point* untuk mendapatkan alamat *IP Statis*, dengan cara *Tathering* menggunakan *smartphone*.
2. Menyambungkan *client* ke *Access Point* untuk mendapatkan *IP*.
3. Melakukan cek koneksi dengan cara “*ping*” ke alamat *IP* yang akan dituju untuk memastikan terhubung satu sama lain.
4. Pada Sistem Operasi *Ubuntu* ,ketika menjalankan *software Mqtt Client* ,*server* akan otomatis hidup tanpa harus mengkonfigurasinya.

3.6 Menguji Server *Mosquito*

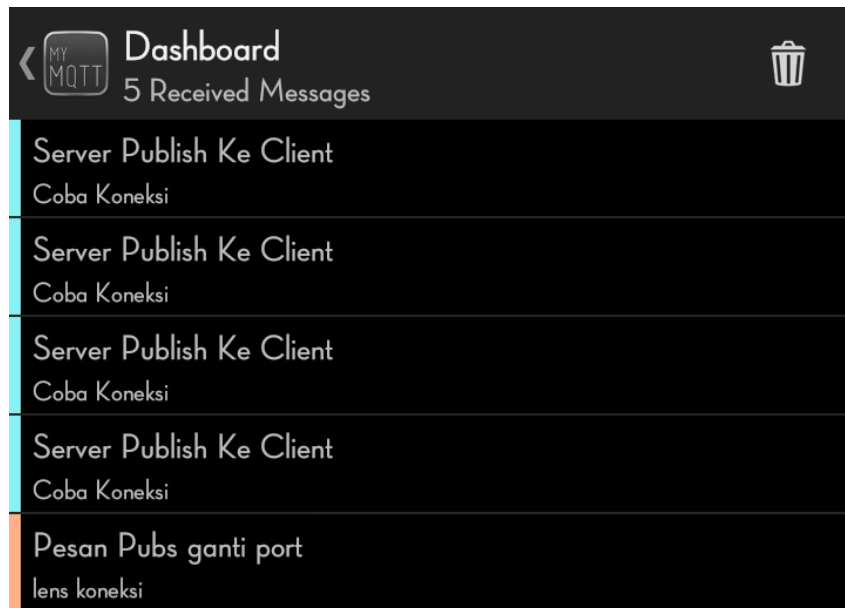
Gambar dibawah ini merupakan hasil dari konfigurasi *server* dan client . Dimana *server* menggunakan port yang sudah dirubah menjadi 1884 kemudian . ada dua buah client yang terdeteksi pada saat *server* sedang digunakan.

```
shiorin@shiorin-team8-B490:~$ mosquitto -p 1884
1469984179: mosquitto version 1.4.9 (build date Fri, 17 Jun 2016 15:43:08 +0100) starting
1469984179: Using default config.
1469984179: Opening ipv4 listen socket on port 1884.
1469984179: Opening ipv6 listen socket on port 1884.
1469984219: New connection from 192.168.43.1 on port 1884.
1469984219: New client connected from 192.168.43.1 as root.1469984218973 (c1, k60).
1469984229: New connection from 192.168.43.52 on port 1884.
1469984230: New client connected from 192.168.43.52 as lens_V7riuVBdFuIoXSTJmKua
64dCkOr (c1, k120).
1469984294: Client lens_V7riuVBdFuIoXSTJmKua64dCkOr disconnected.
1469984322: New connection from 192.168.43.52 on port 1884.
1469984322: New client connected from 192.168.43.52 as lens_fTOCvuh21ZMQyW6RyZ
qmF41wP (c1, k120).
1469984674: Client lens_fTOCvuh21ZMQyW6RyZqmF41wP disconnected.
1469984723: New connection from 192.168.43.52 on port 1884.
1469984723: New client connected from 192.168.43.52 as lens_wTpusSP2UnLzNzrdNAWhmu0NJt9 (c1, k120).
1469985172: Socket error on client root.1469984218973, disconnecting.
1469985608: New connection from 192.168.43.123 on port 1884.
1469985608: New client connected from 192.168.43.123 as lens_EDM3MAaTKDIUX7ZKziT1dG6RXTp (c1, k120).
```

Gambar 3.6 Pengujian Server

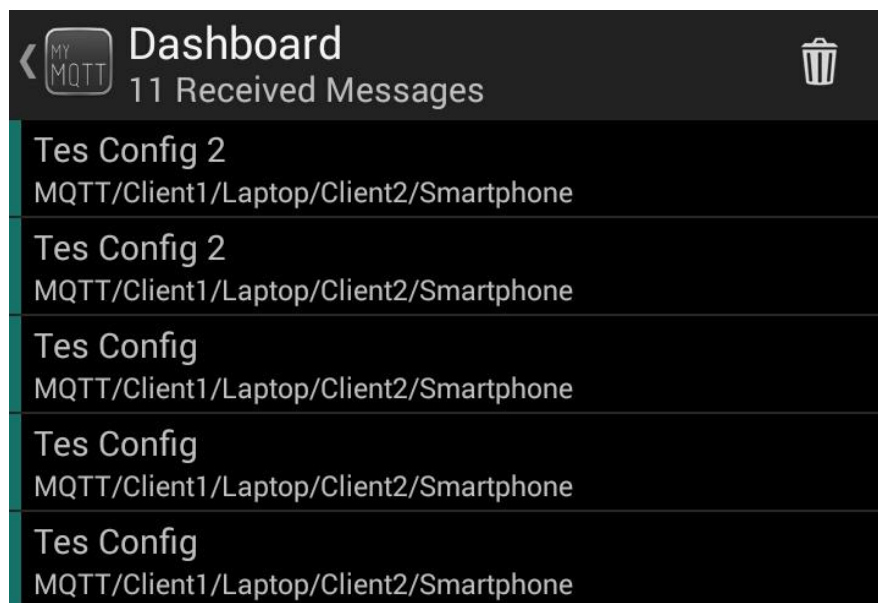
3.6.1 Pengujian Client (*MyMQTT*)

Berikut ini merupakan tampilan pengujian *client* hasil *Publish* dari *server* ke *client* .



Gambar 3.6.1 Hasil Pengujian *MyMQTT*

- Hasil dari pengujian diatas yaitu , *client* menerima pesan dari *server* secara langsung.



Gambar 3.6.1 Hasil Pengujian *Group MyMQTT*

- Gambar diatas merupakan hasil dari ,*subscribe* aplikasi *MyMQTT* ke *server* untuk menerima *Publish*.

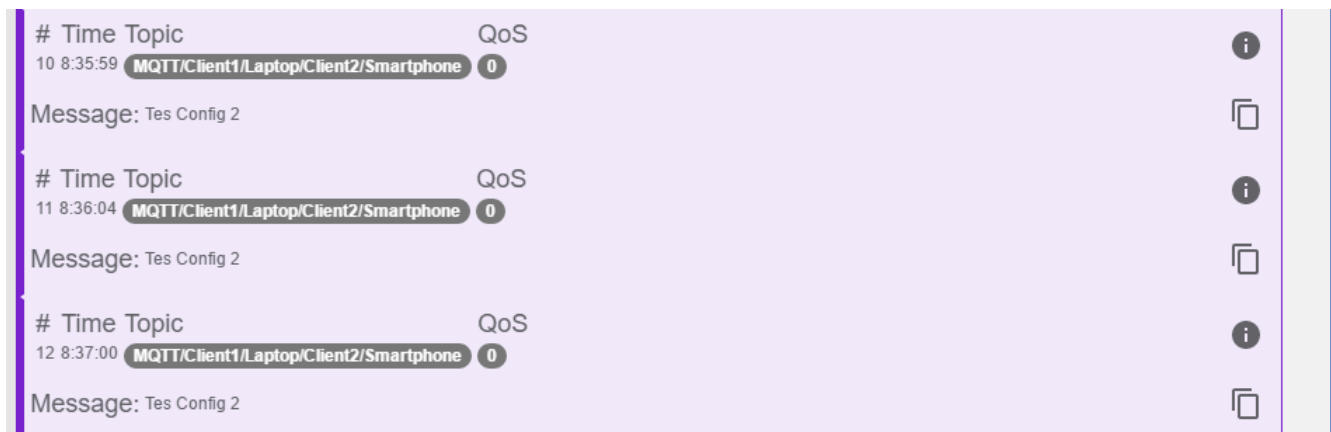
3.6.2 Pengujian Client (*MQTTLens*)

Berikut ini merupakan tampilan pengujian *client* hasil *Publish* dari *server* ke *client* .



Gambar 3.6.2 Hasil pengujian *MQTTLens*

- Menyamakan *Topic* yang dibuat di *server* kemudian *subscribe* ,maka *client* akan langsung menerima respon dari *server*.



Gambar 3.6.2 Hasil pengujian *Group MQTTLens*

Hasil dari konfigurasi Topic dengan menggunakan pengelompokan